2

FORMALIZATION OF TRADEOFF RULES AND OTHER TECHNIQUES
FOR COMPREHENDING COMPLEX RULE-BASED MODELS

Paul K. Davis, Steven C. Bankes, and Robert Weissler

November 1987

P-7394

FORMALIZATION OF TRADEOFF RULES AND OTHER TECHNIQUES FOR COMPREHENDING COMPLEX RULE-BASED MODELS[1]

Paul K. Davis, Steven C. Bankes, and Robert Weissler
The RAND Corporation
1700 Main Street, P.O. Box 2138
Santa Monica, CA 90406-2138

The authors and colleagues have recently developed a large and complex knowledge-based simulation that includes political-military decision models, each with thousands of qualitative rules. As part of our knowledge architecture we have emphasized rule hierarchies, which require recursive combining rules that evaluate a given qualitative variable as a function of several lower-level variables--often in ways that involve value-laden tradeoffs. Recently, we have begun to formalize procedures for defining the qualitative variables and characterizing the tradeoff relationships in algebra-like terms. This has proven valuable in speeding model development and communicating results. Moreover, it has improved the *quality* of rule-writing and has made it easier to differentiate among and deal intuitively with different *types* of combination rules--many of which are quite different from the standard weighted-sum approach. This paper motivates the formal procedures with examples, notes that the problems involved are generic rather than domain specific, and then illustrates an approach for dealing with them. In the longer run, there should be implications for rule algebras in formal modelling and new syntaxes in programming languages. In essence, the objective should be to state combining rules at a high level of abstraction.

INTRODUCTION

The motivation for the current paper was the need to address practical problems arising in the authors' work on a knowledge-based simulation for analytic war gaming called the RAND Strategy Assessment System (RSAS) (see description in Davis, 1986a). The RSAS includes "National Command Level" (NCL) models that make political-military

[1]This paper was presented at the *4th International Symposium on Modelling and Simulation Methodology: Intelligent Environments and Goal-Directed Models*, January 21-23, 1987, co-sponsored by the University of Arizona and The Society for Computer Simulation, held in Tucson, Arizona.

decisions using rules addressing variables deemed important by strategic analysts and/or players in human games (Davis, Bankes, and Kahan, 1986). *Building* the models can be an insightful mechanism for studying deterrence issues (Davis, 1986b), and the models themselves can then be used in both games and simulations.

There are alternative NCL models to represent different plausible national behaviors and grand strategies. The baseline models from which others are relatively simple variants translate into programs with the equivalent of about 45,000 lines of "C" code. Thus, the models are considerably more complex than typical rule-based systems. They are also unusual in addressing *normative* issues involving national values, grand strategy, and other matters on which there are no experts in the usual sense of expert systems (Davis, 1986b).

In attempting to develop NCL models that would "reason" in ways that humans would find comfortable, the authors found themselves confronting problems generic to a broader class of qualitative rule-based models. After providing some general background on the model, we shall discuss and illustrate two such issues in some detail: (1) *minimizing ambiguities when using qualitative variables*, and (2) *formalizing the combining rules necessary when dealing with a hierarchy of variables*.

REPRESENTING KNOWLEDGE IN NCL MODELS

Organizing knowledge for something as complex as political-military decision models is a major challenge. The principal techniques we have used are (Davis, Bankes, and Kahan, 1986):

- A cognitively natural *process model of decision* (Figure 1).
- *Grouping rules by conflict level* within a given process.
- Organizing rules within a group by exploiting *hierarchies of variables* (e.g., Figure 2).
- Expressing groups of related rules with the use of *decision tables*.

Figure 1 describes the process model, which includes: situation assessment; "learning" (more modestly, adjustment of assumptions based on experience in the simulated war); projections using the model's updated assumptions; deciding objectives and strategies; and, finally, establishing special controls on such matters as rules of engagement. As indicated by the last process, the model can then test the implications of a tentatively chosen strategy with both heuristic rules and a look-ahead projection (a game within a game using the NCL model's assumptions about the world). If the test is negative there is feedback and reconsideration. In essence, the NCL model is using simulation as a knowledge source.

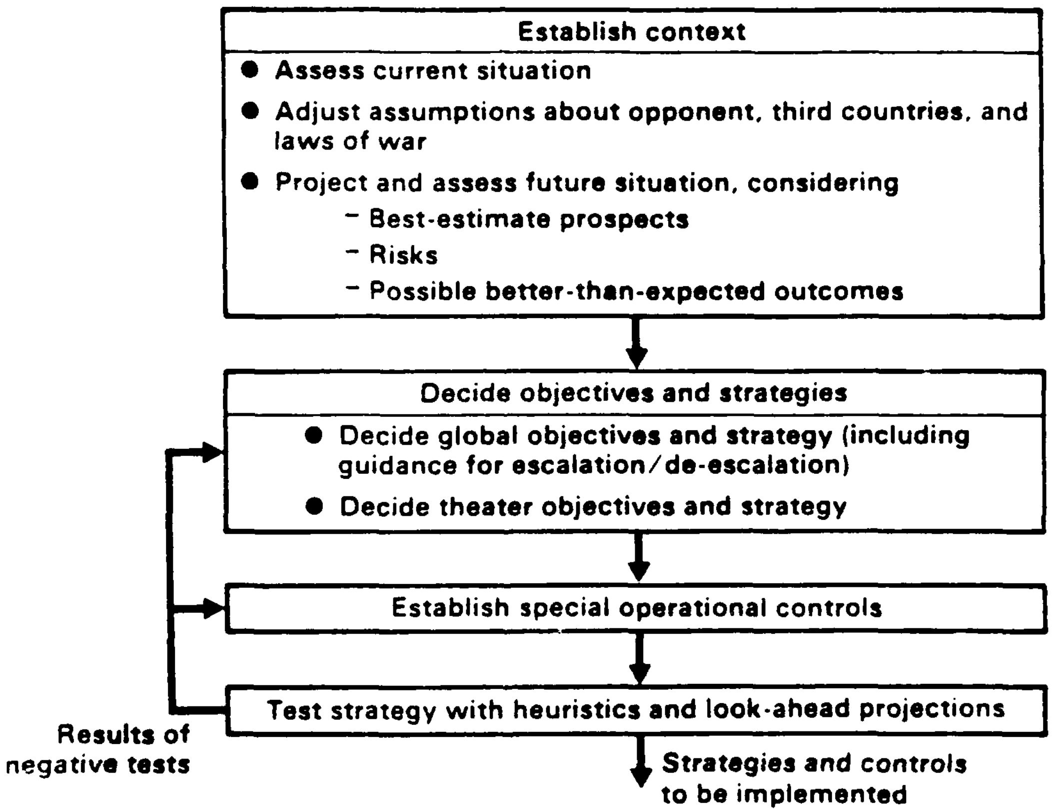
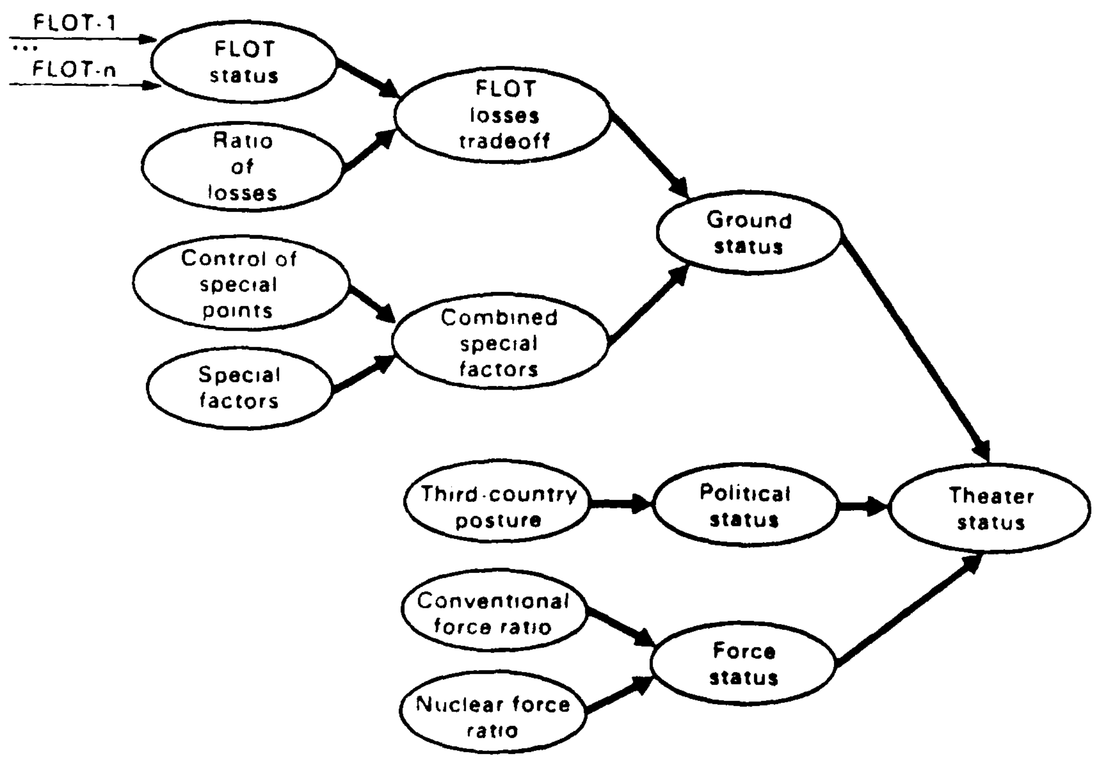


Fig. 1--National command level model
top level program structure

We shall not discuss the second technique ("chunking" rules by conflict level) here, but Figure 2 illustrates the hierarchy-of-variable approach with the variable Theater-status, which depends on the status of the ground war, the state of political alliances, and the state of surviving forces.

Although the hierarchical approach may seem straightforward, it appears from the expert-system literature that many practitioners have not appreciated its importance--especially when building expert systems more as a compilation of rules than as coherent *models* (Davis, 1986a; see also the paper by Clancy in this volume). In practice, having a hierarchy of *intermediate variables* is essential for the understandability of many expert systems. Without such variables,



FLOT = Forward line of own troops.

Fig. 2--An illustrative hierarchy of variables

explanations tend to be verbose--referring to many rules and variables with little focusing and little if any communication of the expert's causal model. There have been several efforts to do better in this regard. Swartout (1983) and Swartout (1981) describe the XPLAIN system, which uses concepts such as "automaticity" to tie together the effects of calcium, potassium, and digitalis in affecting ventricular fibrillation. Buchanann and Shortliffe (1984) discuss the role of "compromised host" in MYCIN. In some expert-system environments, one can control the verbosity of explanations and make additional queries as needed (Waterman et al., 1986).

SELECTED ISSUES: PROBLEMS AND APPROACHES

With this background, let us consider the two classes of problem alluded to earlier: minimizing ambiguity in working with qualitative variables and developing formal concepts for combining such variables.

On Ambiguity in the Use of Qualitative Variables

Consider first a domain-specific example visible in Figure 2. The variable at the top left is FLOT-status. The FLOT is the "forward line of own troops" separating the antagonists; there are different FLOTs for each axis of advance in a theater, as indicated by the numerical data FLOT-1,...FLOT-n. FLOT-status has the range {Very Bad, Bad, Mixed, Good, Very Good}, as do the variables FLOT-losses-tradeoff, Ground-status, and Theater-status. But what do we *mean* by a "Bad" status or a "Good" one? These have no inherent definition, but must instead be given meaning by the rule-writer. To make matters worse, the rule-writer's intuitive meaning may reflect implicit and context-dependent deep knowledge. Perhaps his assessment of FLOT-status depends on an assumed military strategy such as an initial willingness to give ground in exchange for time. Even worse, it may depend on assumptions about what is going on elsewhere in the world and when reinforcements will arrive. Clearly, this type of ambiguity can cause severe problems.

A second type of ambiguity arises when one uses the same variable in more than one context in a structural sense. Suppose the variable FLOT-status is relevant to the evaluation of both Theater-status and another high-level variable, Theater-risks. To say FLOT-status is "Mixed" would convey different intuitive meanings in the two contexts. Why? Consider a commander who has lost ground over a period of ten days. If his losses were no worse than expected given his strategy, then perhaps FLOT-status in the context of evaluating Theater-status would be Mixed (i.e., neither Good nor Bad in view of the circumstances). However, in evaluating Theater-risks--even if he had a concept for recovering ground--he might characterize FLOT-status as Bad because of uncertainty about his actual ability to do so.

This subtle shift of meaning from one context to another is an endemic problem for natural-language rules: Their apparent clarity is quite misleading, and developing formal protections to avoid the ambiguity is a generic challenge.

Those familiar with software engineering practices may immediately argue that the key is modularity: The variables should be defined locally so that they cannot be used outside the context for which they are originally introduced. In our experience, however, this is not so clear-cut. The subject-area specialist may "know" that a given lower-level variable, with precisely the same "meaning," is relevant to two or even many higher-level variables. He may even argue that a particular "lower-level variable" is a prime determinant of a particular higher-level variable. The result may be something like that illustrated in Figure 3, where the dashed lines show flows one would like to avoid.

It might appear that such complications rule out a hierarchical approach because of lattice-like connections, but in our work--and we suspect in many applications--it is possible to salvage the essence of the hierarchical approach by generalizing it as shown in Figure 4, which shows a near hierarchy of variable *groups* with the out-of-hierarchy flow being exceptional. The ambiguity potential, however, may be severe: intuition will say that a given variable is a determinant of several higher level variables, but a closer look may indicate different shades

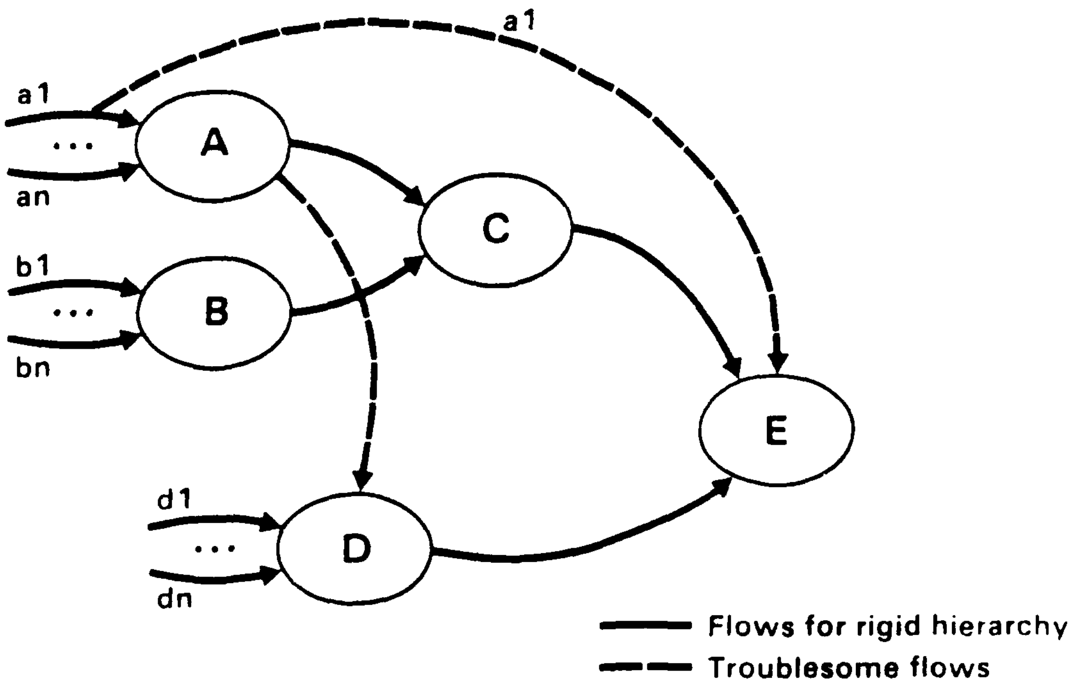


Fig. 3--Example decision table for setting overall objective

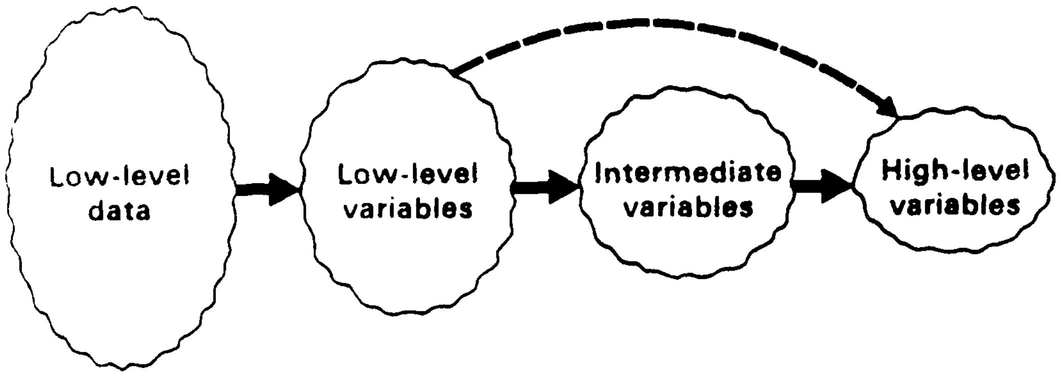


Fig. 4--A hierarchy of variable groups

of meaning associated with its qualitative values in the different contexts.

What, then, do we propose? This is *not* a case where stating the problem defines the solution. We are chary about stating general guidelines, because knowledge-based systems should be comfortable to those providing the knowledge and/or to those using the systems, even if they make life more difficult for the builder of software. Nonetheless, we recommend tilting in the direction of modularity and software engineering because, in our experience, the intuitive aspects of knowledge--especially in dealing with value-laden qualitative variables--have proven unreliable: Domain specialists for whom a certain structure and semantics is self-evident one day themselves have trouble with the ambiguities when they revisit the problem on another occasion. Our suggested principles are:

- *Define the meaning of qualitative variables as they are introduced*, with examples to bring out distinctions and subtleties (in our work, this corresponds to adding comments to the RAND-ABEL data-dictionary items). Reject claims that natural-language rules are intuitive.

- *Plan from the start to parameterize the meanings* (e.g., so whether Ground-status is "Good" or "Bad" depends on such items as strategy). The parameter values may change dynamically.

- Where possible, *rule out having ambiguous variables affect more than one higher-level variable*. Where this would sacrifice important elements of the conceptual model's natural language, consider notational techniques to emphasize that different variables are related (e.g., Ground-status1, Ground-status2, and so on). Also, use programming techniques to isolate where the variable is evaluated (e.g., turn the variable into a function to prevent dispersal of assignment statements).

- *Attempt to persuade domain specialists to restructure their rules so that lowest-level variables do not appear in the determination of highest-level variables*. (Example from Figure 2: The location of the FLOT on axis 2, FLOT-2, might be especially important in determining Theater-status, but that can probably be accounted for by weighting the effect of FLOT-2 in the evaluation of the intermediate variable FLOT-status). Recognize, however, that this is sometimes counterproductive because it may substitute multilevel abstractions for a straightforward heuristic.

- Where lattice-like relationships are unavoidable, *identify and exploit conceptual aggregations* of variables such as those suggested in Figure 4.

Toward an Algebra of Tradeoff Rules

Let us next consider the issue of combining rules. Figures 2-4 all describe knowledge bases in which a given high-level variable is determined by lower-level variables. But what are the rules for making these determinations? Figure 5 illustrates combining rules for the NCL model's evaluation of Theater-status, consistent with the hierarchy of Figure 2. A simplified version of actual RAND-ABEL™ source code (see Shapiro, Hall, Anderson, and LaCasse, 1985, for discussion of RAND-ABEL™), consists of a decision table, the first line of which is read "*If* Ground-status is Very-good and Political-status is greater than Bad and Force-status is greater than Bad, *Then* Theater-status is Very-good." This structure exploits the natural ordering of the range {Very-bad, Bad, Mixed, Good, Very-good}, so that >Bad really means (Mixed or Good or Very-good). It enables us to express 600 If-Then-Else cases in 13 lines, of which only the first six are shown here.

Figure 5 shows *how* we combine qualitative variables with subjectively determined values, but it provides no *rationale*.

```
Decision Table

Ground-    Political-   Force-        /  Theater-
status     status       status        /  status
========= ========== ===========   /  ==========.
Very-good >Bad          >Bad            Very-good
Very-good  --            --             Good
Good       Very-good    Very-good       Very-good
Good       >Very-bad    >Very-bad       Good
Good        --           --             Mixed
Mixed      Very-good    Very-good       Good
[other lines deleted here for brevity]
[End Table].
```

Fig. 5--Illustrative decision table for setting Theater-status

One might be inclined to write rules of this sort by translating the qualitative values into integers, summing, dividing, and then translating back into qualitative values. However, that familiar approach is quite inadequate for representing many of the logical relationships we have observed in our knowledge base. Nor is it sufficient to supplement the approach by weighting some of the variables more than others.

If we fill out decision tables working directly with the qualitative concepts and attempting to use all our domain-specific knowledge, and if we then review the resulting decision tables, we find a considerable diversity of combining rules at work. As mentioned, alternative NCL models represent different national temperaments and grand strategies. One model may tend toward conservatism in its situation assessments: If *any* of the contributors to Theater-status is Very-bad, then its overall assessment may be Very-bad, regardless of the other factors. A different model may trade the variables off as a weighted sum. Yet another model may tend toward optimism--emphasizing the favorable reports and discounting the unfavorable ones. While such behaviors might seem irrational to an operations researcher, they *are* familiar in the real world. One reason for constructing alternative models is to highlight such matters so that human teams using the models as decision aids in educational games can consider whether they "want" to be optimistic, conservative, or neutral.

Because there is reason for a diversity of combining rules, it is reasonable to consider formalizing them. *The approach we suggest has three steps*:

- Develop a list of alternative combination-rule concepts.
- Define each concept rigorously.
- Develop a high-level algebraic representation or a language syntax.

Based on the patterns of rules observed in our work, we would mention at least the following combination-rule concepts, which the reader may wish to think about in terms of human styles he has observed himself in tradeoff decisions:

- Simple averaging (e.g., Good + Bad = Mixed)
- Rounding-up averaging (e.g., Good + Mixed = Good)
- Rounding-down averaging (e.g., Good + Mixed = Mixed)
- Reinforcement-averaging (round toward the closer end)
- Damped-averaging (round toward the center)
- Marginal addition (e.g., Good + Good = Very Good, but Bad + Mixed = Bad)
- Minimum-of (worst of)
- Maximum-of (best of)

Defining the concepts amounts to writing down generic decision tables. Formalizing the concepts in a modelling algebra or a programming language requires taking additional steps such as implementing in RAND-ABEL™ syntaxes, for example:

> Let C be the report from simple-averaging-roundup using A as first-variable and B as second variable.

This assignment statement for a qualitative variable such as Ground-status would be equivalent to a decision table, since that table would be part of the definition of a RAND-ABEL™ function *simple-averaging-roundup*. We should mention in passing that this approach requires that the variables to be combined are similarly *calibrated* so that such values as "Bad" have the same weight. Also, some of the combining rules make sense only if the values are in some sense "equally spaced."

Figure 6 shows a notional example of a tradeoff relationship in which an attacking commander is trying to judge his net status taking into account his FLOT status and a measure of relative attrition. The top portion of the figure is a comment in the form of a matrix "picture" of the tradeoff. An entry in the matrix gives the value of the tradeoff. The bottom part of the figure is executable code. The matrix is not symmetric because of some deep knowledge: For example, even if the commander's attrition has been low relative to his opponent's, a failure to take ground according to plan is not acceptable; however, if he has taken considerable ground, then that can compensate for considerable attrition. An algebra describing this combining rule would be noncommutative.

Although we have not yet pursued the formalization of these concepts very far, we have nonetheless found it fruitful to itemize and give names to the combining rules listed above. First of all, it has simplified communication between the rule-writers and programmers: The rule-writer may say nothing more complicated than "combine these variables with simple averaging and rounding upward"; he need not fill out a complete decision table. Second, comment to this effect is very useful as documentation. Third, being sensitive to the issue has affected validation reviews. Not uncommonly, we revise decision tables with such comments as: "In this case, I can't really justify the lack of symmetry in my decision table. I don't remember what I had in mind, but it seems I must have been thinking fuzzily. Let's go instead with ...(the name of some combining rule)." In other cases, we *insert* asymmetries to reflect particular human-like reasoning. In future work, we hope to use decision models as a mechanism for discussing interesting failures of human decisionmakers such as their tendency to give unreasonably large (or small) weights to the last information received--or, more generally, to make different decisions depending on the order in which they process information.

[

Attacker FLOT status					
VG	M	G	VG	VG	VG
G	B	M	G	G	VG
M	VB	B	M	M	G
B	VB	VB	B	B	M
VB	VB	VB	VB	VB	B
	VB	B	M	G	VG

Attacker Measure of Relative Attrition

Tradeoff Table for FLOT-status and Relative-attrition

]

Decision Table [Tradeoff Table for Attacker]

flot-status	relative-attrition /	flot-losses-tradeoff
Very-good	>=Mixed	Very-good
Very-good	Bad	Good
Very-good	Very-bad	Mixed
Good	Very-good	Very-good
Good	Good	Good
Good	Mixed	Good
Good	Bad	Mixed

[Lines deleted for brevity]

Fig. 6--Illustrative decision table and pictorial comment

SUMMARY

In summary, we have drawn upon our experience in one complex model-building effort to suggest a set of guidelines that may prove useful to others because the underlying problems are generic to hierarchies of natural-language rules with qualitative and subjective values. We have suggested that some of the ideas could be incorporated in a formal rule-algebra for modelling or in the syntaxes of programming languages to

permit workers to describe at a higher level, and thus more transparently, the nature of the combining rules used in moving up a hierarchy of variables.

BIBLIOGRAPHY

Buchanan, Bruce G., and Edward H. Shortliffe (eds.), (1984), *Rule-Based Expert Systems: The MYCIN Experiments of the Heuristic Programming Project*, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts.

Davis, Paul K. (1986a), "Applying Artificial Intelligence Techniques to Strategic-Level Gaming and Simulation," in M. S. Elzas, T. I. Oren, and B. P. Zeigler (eds.), *Modelling and Simulation Methodology in the Artificial Intelligence Era*, Elsevier Science Publishers, North Holland. Also available as (same title), The RAND Corporation, P-7120, November 1985.

Davis, Paul K. (1986b), "A New Analytic Technique for the Study of Deterrence, Escalation Control, and War Termination," in Stephen J. Cimbala (ed.), *Artificial Intelligence and National Security*, Lexington Books, D. C. Heath and Company, Lexington, MA.

Davis, Paul K., Steven C. Bankes, and James P. Kahan (1986), *A New Methodology for Modeling National Command Level Decisionmaking ir War Games and Simulations*, The RAND Corporation, R-3290-NA.

Shapiro, Norman Z., H. Edward Hall, Robert H. Anderson, and Mark LaCasse (1985), *The RAND-ABEL™ Programming Language: History, Rationale, and Design*, The RAND Corporation, R-3274-NA.

Swartout, William R. (1981), "Explaining and Justifying Expert Consulting Programs," *Proceedings of the 7th International Joint Conference on Artificial Intelligence*.

Swartout, William R. (1983), "Explain: A System for Creating and Explaining Expert Consulting Programs," *Artificial Intelligence, 21*, pp. 285-325.

Waterman, D. A., J. Paul, B. A. Florman, and J. R. Kipps (1986), *An Explanation Facility for the ROSIE Knowledge Engineering Language*, The RAND Corporation, R-3406-ARPA/RC.